LSTM(Long Short-Term Memory)_ 순환 신경망

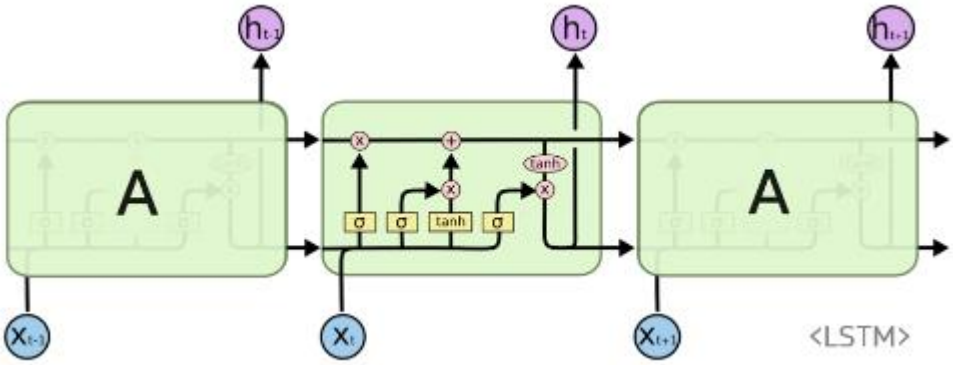
: RNN의 한 종류이자 RNN의 장기 의존성 문제를 해결 + Graidient Vanishing/ Exploding 개선하기 위해 고안된 architecture

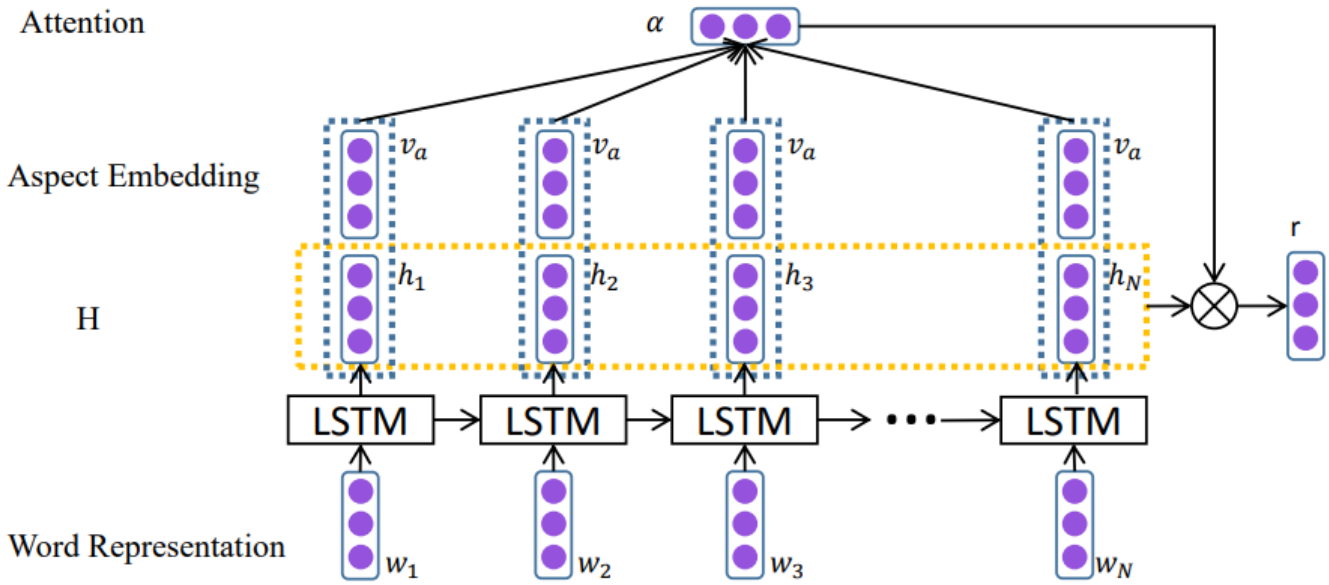
Gradient Vanishing (기울기 소실)

-> 학습 과정에서 모델이 깊어지거나 장기 의존성을 갖는 데이터 학습시 층을 거치면서 기울기를 곱하는 과정에서 크기가 작아져 소멸하는 현상

Gradient Exploding (기울기 폭주)

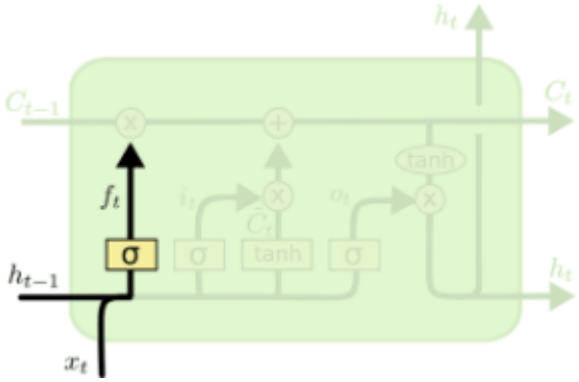
-> 가중치 업데이트 이후에 학습 과정에서 수렴속도가 늦어지거나 혹은 발산되어 학습 중단





<Gate mechanism>

1. f(t) (forget gate)



$$f_t = \sigma(W_f \cdot X + b_f)$$

이전의 정보 잊거나 기억하는 과정

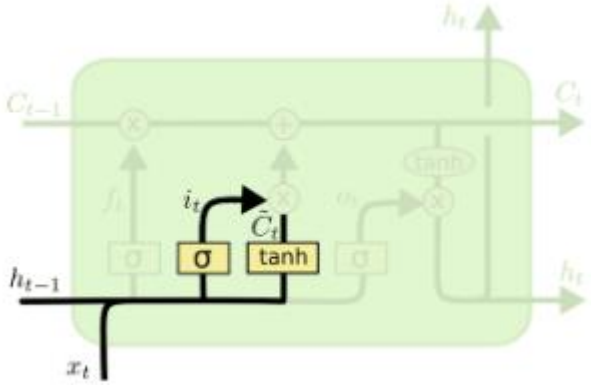
σ를 통하여 previous state 얼마나 기억할지 결정.

Sigmoid -> 0 : 이전 상태 지움 1 : 이전 상태 기억

즉, 각 정보의 상태가 얼마나 중요한지를 나타낸다고 생각하면 쉽게 이해가 가능하다.

```

# Python

Import tensorflow as tf

# inputs:입력 값, inputs_weight:입력된 값에 부여할 가중치, biases: 편향

Forget_gate = tf.sigmoid(tf.matmul(inputs, inputs_weight) + biases )

```

2. i(t) (input gate)



$$i_t = \sigma(W_i \cdot X + b_i)$$

σ를 통하여 새로운 값(g(t)_현재 정보 기억) 얼마나 반영할지 결정

같은 입력으로 tanh로 연산을 한 값을 내보내는 값

현재 단계의 입력과 이전 단계의 은닉 상태를 기반으로 어떤 정보를 추가할지 결정

Activation 함수를 Sigmoid(0~1사이의 값으로 제한)로 설정해 추가할 비율을 설정합니다.

```

# Python

Import tensorflow as tf

# inputs:입력 값, inputs_weight:입력된 값에 부여할 가중치, biases: 편향

input_gate = tf.sigmoid(tf.matmul(inputs, input_weights) + biases)

```

3. Cell state

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c)$$

-> LSTM의 셀 상태 확인

```

# python

# tanh -1 ~ 1사이 값으로 확인

#c(t-1) : 이전 셀 상태 정보

cell_state = tf.matmul(Forget_gate, c(t-1)) + tf.tanh(tf.matmul(inputs, weights) + weights)

```

+ return_sequences = True 관한 추가 설명

1. time step 예측 : 각 단계별 time step의 값을 예측 할 수 있다.

2. debugging을 통해 모델의 작동과정 확인 가능

3. 출력 값 조절로 다양하게 활용 가능 -> 제가 말했던 LSTM으로 텍스트 값 추출할 수 있을 것 같다고 말한 부분입니다~

```
class LstmModel:
    def __init__(self):
        self.model = self.build_model()
    def build_model(self):
        model = Sequential()
        model.add(LSTM(64, input_shape=(24, 100), return_sequences=True))
        model.add(LSTM(32))
        model.add(Dense(16, activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(1, activation='sigmoid'))
        model.compile(loss='binary_crossentropy', optimizer=Adam(), metrics=['accuracy'])
        return model
```

제가 실수한 부분이 activation함수로 relu를 설정한 부분인데 장기의존성 및 기울기 소실이 개선된 LSTM에선 사용할 필요가 없다고 하네요?.....

Relu -> tanh로 바꿔서 실행 해볼게요///